CSE 515: Statistical Methods in Computer Science

Homework #4

Due at noon on March 11th

Guidelines: You can brainstorm with others, but please solve the problems and write up the answers by yourself. You may use textbooks (Koller & Friedman, Russel & Norvig, etc.) and lecture notes from class. Please do NOT use any other resources or references (e.g., example code, online problem solutions, etc.) without asking.

Submission instructions: Submit this assignment by email to Chloé Kiddon (chloe@cs). Attachments should include: A PDF containing written answers. Typed answers are **highly** preferred, but if this is a *hardship*, then handwritten answers are fine as long as they are **completely** legible.

1. Consider learning the following Bayesian network: $A \rightarrow B \rightarrow C$. And the following data table, with entries '?1' and '?2' missing at random:

A	B	C
T	?1	T
F	T	F
F	F	F
T	F	T
F	F	?2
T	T	T

   (a) Use the data to estimate initial parameters for this network, using maximum likelihood estimation for simplicity.

   (b) Perform two iterations of the EM algorithm (by hand) to estimate the values of the missing data, reestimate the parameters, reestimate the values of the missing data, and reestimate the parameters once more. Show your calculations.

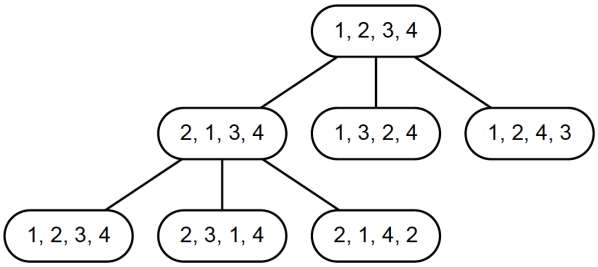

Figure 1: Partial search tree example for orderings over variables X_1, X_2, X_3, X_4. Successors to $\prec = (1, 2, 3, 4)$ and $\prec' = (2, 1, 3, 4)$ shown.

2. Consider learning the structure of a Bayesian network for some given ordering, $\prec$, of the variables $X_1, \ldots, X_n$. (This can be done efficiently as described in section 18.5.2.1 of the textbook.) Now assume that we want to perform a search over the space of orderings; that is, we are searching for a network (with bounded in-degree k) that has the highest score. We do this by defining the score of an ordering as the score of the (bounded in-degree) network with the maximum score consistent with that ordering, and then we search for the ordering with the highest score. We bound the in-degree so that we have a smaller and smoother search space.

   We will define our search operator, o, to be "Swap X_i and X_{i+1}" for some $i = 1, \ldots, n-1$. Starting from some given ordering, $\prec$, we evaluate a decomposable structure score of all successor orderings, $\prec'$, where a successor ordering is found by applying o to $\prec$ (see Figure 1). We now choose a particular successor, $\prec'$. Provide an algorithm for computing *as efficiently as possible* the score for all successors of the new ordering, $\prec'$, given that we have already computed the scores for all successors of $\prec$.

   Note: A structure score $\text{score}(\mathcal{G} : \mathcal{D})$ is *decomposable* if the score of a structure $\mathcal{G}$ can be written as

$$\text{score}(\mathcal{G} : \mathcal{D}) = \sum_i \text{FamScore}(X_i | \text{Pa}_{X_i}^{\mathcal{G}} : \mathcal{D}),$$

   where the *family score* $\text{FamScore}(X|\mathbf{U} : \mathcal{D})$ is a score measuring how well a set of variables $\mathbf{U}$ serves as parents of X in the data set $\mathcal{D}$.

3. Show that adding edges to a Bayesian network never decreases the likelihood.

4. Naive Bayes (NB) and logistic regression (LR) have the same form, but naive Bayes is a generative model (learned using maximum likelihood, to maximize $P(x, y)$), while logistic regression is a discriminative model (learned using maximum conditional likelihood, to maximize $P(y|x)$). In this problem, assume both models are learned on the same training data with no prior. The conditional log likelihood (CLL) on a dataset D is defined as:

$$\sum_{(x,y) \in D} \log P(y|x).$$

(a) Is it possible for NB to have a higher CLL than LR on the training data? If so, under what conditions will this be true? If not, explain why not.

(b) Is it possible for NB to have a higher CLL than LR on separate testing data? If so, under what conditions will this be true? If not, explain why not.

5. In class, we discussed the problem of estimating the parameters of both Markov networks and Bayesian networks using Maximum Likelihood Estimation (MLE). In this problem, we consider MLE for the parameter estimation of Conditional Random Fields. A Conditional Random Field (CRF) encodes the following distribution:

$$P(\mathbf{Y}|\mathbf{X}) = \frac{1}{Z(\mathbf{X})} P'(\mathbf{Y}, \mathbf{X}).$$

In this problem, we will consider the log-linear parameterization of a CRF, so that the network is annotated with a set of n features $f_i[\mathbf{X}_i, \mathbf{Y}_i]$, where $\mathbf{Y}_i \neq \emptyset$, and weights w_i. Thus we have:

$$P'_{\mathbf{w}}(\mathbf{Y}, \mathbf{X}) = \prod_{i=1}^{n} \exp(w_i f_i(\mathbf{X}_i, \mathbf{Y}_i))$$

and

$$Z(\mathbf{X}) = \sum_{\mathbf{Y}} P'_{\mathbf{w}}(\mathbf{Y}, \mathbf{X}).$$

We know that the derivative of the log-likelihood for a standard log-linear Markov network is

$$\frac{\partial}{\partial w_i} \ell(\mathcal{D} : \mathbf{w}) = (\mathbf{E}_{\hat{P}}[f_i] - \mathbf{E}_{P_{\mathbf{w}}}[f_i]),$$

where $\mathbf{E}_{\hat{P}}[f_i]$ is the empirical expectation of f_i in the dataset and $\mathbf{E}_{P_{\mathbf{w}}}[f_i])$ is the expectation of f_i in our model parameterized by $\mathbf{w}$.

Now we come to the questions. In the following, you should assume you are given a dataset $\mathcal{D} = \{\langle \mathbf{x}[1], \mathbf{y}[1] \rangle, \ldots, \langle \mathbf{x}[M], \mathbf{y}[M] \rangle\}$.

(a) Write the log-likelihood $\ell(\mathcal{D} : \mathbf{w})$ for a log-linear CRF $\mathcal{C}$.

(b) Prove that the derivative of $\ell(\mathcal{D} : \mathbf{w})$ with respect to w_i is the following:

$$\frac{\partial}{\partial w_i} \ell(\mathcal{D} : \mathbf{w}) = \sum_{m=1}^{M} f_i(\mathbf{x}_i[m], \mathbf{y}_i[m]) - \mathbf{E}_{(\mathbf{Y}_i|\mathbf{x}[m]) \sim P_{\mathbf{w}}}[f_i(\mathbf{x}_i[m], \mathbf{Y}_i)],$$

where $\mathbf{E}_{(\mathbf{Y}_i|\mathbf{x}[m]) \sim P_{\mathbf{w}}}[f_i(\mathbf{x}_i[m], \mathbf{Y}_i)]$ is the expectation of f_i given $\mathbf{x}[m]$ in our CRF with the distribution $P_{\mathbf{w}}$. That is,

$$\mathbf{E}_{(\mathbf{Y}_i|\mathbf{x}[m]) \sim P_{\mathbf{w}}}[f_i(\mathbf{x}_i[m], \mathbf{Y}_i)] = \sum_{\mathbf{Y}_i} f_i(\mathbf{x}_i[m], \mathbf{y}_i) P_{\mathbf{w}}(\mathbf{y}_i|\mathbf{x}[m]).$$

(c) Given the above derivative, why is learning a CRF computationally more expensive than learning a standard (generatively trained) Markov network?